

Claims:

1 1. A method for operating a server computer as part of a group of server computers to
2 maintain session states for a plurality of users, the method comprising:
3 creating a first session state and locally storing the session state;
4 transmitting a command to the group of server computers that, when executed by an
5 executing server computer of the group of server computers, causes the executing server computer
6 to locally store the first session state;
7 receiving a command from another server computer of the group of server computers that
8 corresponds to a second session state; and
9 executing, by the server computer, the command to create the second session state and to
10 locally store the second session state.

1 2. The method of claim 1, further comprising:
2 receiving a request for a third session state from a locally coupled process;
3 determining that the third session state is not locally stored; and
4 requesting the third session state from at least one other server computer.

1 3. The method of claim 2, further comprising:
2 identifying a primary owner and a secondary owner of the third session state; and
3 transmitting a request for the third session state to the primary owner and the secondary
4 owner of the session state.

1 4. The method of claim 2, wherein the request for the third session state is broadcast to

2 all other server computers of the group of server computers.

1

1 5. The method of claim 1, further comprising:

2 receiving a request from a coupled process to modify the first session state;

3 modifying a local copy of the first session state; and

4 broadcasting a command to other server computers of the group of server computers that,

5 when executed by an executing server computer, causes the executing server computer to locally

6 modify the first session state.

1

1 6. The method of claim 5, wherein modifying the local copy of the first session state

2 includes:

3 deleting the local copy of the first session state; and

4 creating a new copy of the first session state with modifications included therein.

5 7. The method of claim 1, wherein the local copy of the first session state is stored in
6 dynamic memory of the server computer.

1 8. The method of claim 1, wherein the local copy of the first session state is stored in
2 static memory of the server computer.

1

1 9. The method of claim 1, further comprising:

2 deleting the local copy of the first session state; and

3 creating a new copy of the first session state with modifications included therein.

1

1 10. The method of claim 1, further comprising:
2 deleting the local copy of the first session state; and
3 broadcasting a command to other server computers of the group of server computers that,
4 when executed by an executing server computer, causes the executing server computer to locally
5 delete the first session state.

1
1 11. The method of claim 1, further comprising:
2 renewing the local copy of the first session state; and
3 broadcasting a command to other server computers of the group of server computers that,
4 when executed by an executing server computer, causes the executing server computer to locally
5 renew the first session state.

1
1 12. The method of claim 1, further comprising:
2 receiving a request for a requested session state from a requesting server computer of the
3 group of server computers;
4 accessing a local copy of the requested session state; and
5 transmitting a copy of the requested session state to the requesting server computer.

1
1 13. The method of claim 12, wherein:
2 the request is sent to the primary owner and secondary owner of the requested session state;
3 and
4 the server computer is the primary owner or secondary owner of the requested session state.

1 14. The method of claim 12, wherein the request is sent to all server computers of the
2 group of server computers.

1
1 15. The method of claim 1, further comprising identifying at least one other server
2 computer of the group of server computers.

1
1 16. The method of claim 1, further comprising identifying a secondary owner of the
2 other server computers of the group of server computers.

1
1 17. The method of claim 1, further comprising publishing a plurality of locally stored
2 session states to other server computers of the group of server computers.

1
2 18. The method of claim 17, wherein the server computer coordinates the session states
3 it publishes with session states published by other server computers of the group of server
computers.

1 19. A system for operating a server computer as part of a group of server computers to
2 maintain session states for a plurality of users, the system comprising:

3 a fact array that locally stores a plurality of session states on the server computer;

4 a server computer interface thread that couples the fact array, to at least one local and to a
5 broadcast port, wherein the server computer interface thread creates a first session state and locally
6 stores the session state in the fact array;

7 a command publisher thread coupled to the server computer interface thread and to the
8 broadcast port that transmits a command to the group of server computers that, when executed by an
9 executing server computer of the group of server computers, causes the executing server computer
10 to locally store the first session state;

11 a subscriber thread coupled to the fact array, the server computer interface thread, the
12 command publisher thread, and the broadcast port, wherein the subscriber thread receives a
13 command from another server computer of the group of server computers that corresponds to a
14 second session state; and

15 wherein the subscriber thread executes the command to create the second session state and
16 to locally store the second session state in the fact array.

1 20. The system of claim 19, wherein:

2 the command publisher thread transmits a request to determine whether the server computer
3 is the only member of the group of server computers; and

4 the subscriber thread receives at least one response to determine whether the server
5 computer is the only member of the group of server computers.

1 21. The system of claim 19, wherein:

2 the subscriber thread further determines a secondary owner of the group of server
3 computers; and

4 the subscriber thread notifies the command publisher thread of the identity of the secondary
5 owner.

1 22. The system of claim 19, wherein:

2 the server computer interface thread receives a request for a third session state from a locally
3 coupled process;

4 the server computer interface thread determines that the third session state is not locally
5 stored; and

6 the server computer interface thread requests the third session state from at least one other
7 server computer.

1 23. The system of claim 22, wherein the server computer interface thread transmits the
2 request for the third session state to the primary and secondary owners of the third session state.

1 24. The system of claim 22, wherein the server computer interface thread broadcasts the
2 request for the third session state to all other server computers of the group of server computers.

1 25. The system of claim 19, wherein:

2 the subscriber interface thread receives a request from a coupled process to modify the first
3 session state;

4 the subscriber interface thread modifies a local copy of the first session state stored in the
5 fact array; and

6 the command publisher thread broadcasts a command to other server computers of the group
7 of server computers that, when executed by an executing server computer, causes the executing
8 server computer to locally modify the first session state.

1
1 26. The system of claim 25, wherein in modifying the local copy of the first session
2 state:

3 the server computer interface thread deletes the local copy of the first session state from the
4 fact array; and

5 the server computer interface thread creates a new copy of the first session state with
6 modifications included therein in the fact array.

1
1 27. The system of claim 19, wherein the fact array is at least partially created in dynamic
2 memory of the server computer.

1
1 28. The system of claim 19, wherein the fact array is at least partially created in static
2 memory of the server computer.

1
1 29. The system of claim 19, wherein:
2 the server computer interface thread deletes the local copy of the first session state from the
3 fact array; and
4 the server computer interface thread creates a new copy of the first session state in the fact

array with modifications included therein.

30. The system of claim 19, wherein:

the server computer interface thread deletes the local copy of the first session state from the fact array; and

the command publisher thread broadcasts a command to other server computers of the group of server computers that, when executed by an executing server computer, causes the executing server computer to locally delete the first session state.

31. The system of claim 19, wherein:

the server computer interface thread renews the local copy of the first session state stored in the fact array; and

the command publisher thread broadcasts a command to other server computers of the group of server computers that, when executed by an executing server computer, causes the executing server computer to locally renew the first session state.

32. The system of claim 19, wherein:

the subscriber interface thread receives a request for a requested session state from a requesting server computer of the group of server computers;

the command publisher thread accesses a local copy of the requested session state stored in the fact array; and

the command publisher thread transmits a copy of the requested session state to the requesting server computer.

1 33. The system of claim 32, wherein:

2 the request is sent to the primary owner and secondary owner of the requested session state;

3 and

4 the server computer is the primary owner or secondary owner of the requested session state.

1 34. The system of claim 32, wherein the request is sent to all server computers of the

2 group of server computers.

1 35. The system of claim 19, wherein the subscriber thread identifies at least one other

2 server computer of the group of server computers.

1 36. The system of claim 19, wherein the subscriber thread identifies a secondary owner

2 of the other server computers of the group of server computers.

1 37. The system of claim 19, further comprising a fact publisher thread coupled to the

2 fact array that publishes a plurality of locally stored session states to other server computers of the

3 group of server computers.

1 38. The system of claim 37, wherein the fact publisher thread coordinates the session

2 states it publishes with session states published by other server computers of the group of server

3 computers.

1 39. A server computer operated as part of a group of server computers to maintain
2 session states for a plurality of users, the server computer comprising:

3 a processor coupled to a processor bus;

4 dynamic memory coupled to the processor via the processor bus;

5 static memory coupled to the processor via the processor bus

6 an interface coupled to the processor bus that interfaces the server computer to a computer
7 network; and

8 the memory storing a set of instructions executable by the processor, the set of instructions
9 comprising:

10 a plurality of instructions that, upon execution by the processor, cause the server
11 computer to create a first session state and locally store the session state;

12 a plurality of instructions that, upon execution by the processor, cause the server
13 computer to transmit a command to the group of server computers that, when executed by an
14 executing server computer of the group of server computers, causes the executing server computer
15 to locally store the first session state;

16 a plurality of instructions that, upon execution by the processor, cause the server
17 computer to receive a command from another server computer of the group of server computers
18 that corresponds to a second session state; and

19 a plurality of instructions that, upon execution by the processor, cause the server
20 computer to create the second session state and to locally store the second session state.

1

1

1 40. The server computer of claim 39, wherein the set of instructions further comprise:
2 a plurality of instructions that, upon execution by the processor, cause the server computer
3 to receive a request for a third session state from a locally coupled process;
4 a plurality of instructions that, upon execution by the processor, cause the server computer
5 to determine that the third session state is not locally stored; and
6 a plurality of instructions that, upon execution by the processor, cause the server computer
7 to request the third session state from at least one other server computer.

1 41. The server computer of claim 40, further comprising:
2 a plurality of instructions that, upon execution by the processor, cause the server computer
3 to identify a primary owner and a secondary owner of the third session state; and
4 a plurality of instructions that, upon execution by the processor, cause the server computer
5 to transmit a request for the third session state to the primary owner and the secondary owner of the
6 session state.

1 42. The server computer of claim 40, wherein the server computer broadcasts the
2 request for the third session state to all other server computers of the group of server computers.

1 43. The server computer of claim 39, wherein the set of instructions further comprise:
2 a plurality of instructions that, upon execution by the processor, cause the server computer
3 to receive a request from a coupled process to modify the first session state;
4 a plurality of instructions that, upon execution by the processor, cause the server computer
5 to modify a local copy of the first session state; and

6 a plurality of instructions that, upon execution by the processor, cause the server computer
7 to broadcast a command to other server computers of the group of server computers that, when
8 executed by an executing server computer, causes the executing server computer to locally modify
9 the first session state.

1
1 44. The server computer of claim 43, wherein modifying the local copy of the first
2 session state includes:

3 deleting the local copy of the first session state; and

4 creating a new copy of the first session state with modifications included therein.

1
1 45. The server computer of claim 39, wherein the local copy of the first session state is
2 stored in the dynamic memory of the server computer.

1
1 46. The server computer of claim 39, wherein the local copy of the first session state is
2 stored in the static memory of the server computer.

1
1 47. The server computer of claim 39, wherein the set of instructions further comprise:
2 a plurality of instructions that, upon execution by the processor, cause the server computer
3 to delete the local copy of the first session state; and
4 a plurality of instructions that, upon execution by the processor, cause the server computer
5 to create a new copy of the first session state with modifications included therein.

1 48. The server computer of claim 39, wherein the set of instructions further comprise:
2 a plurality of instructions that, upon execution by the processor, cause the server computer
3 to delete the local copy of the first session state; and
4 a plurality of instructions that, upon execution by the processor, cause the server computer
5 to broadcast a command to other server computers of the group of server computers that, when
6 executed by an executing server computer, causes the executing server computer to locally delete
7 the first session state.

1 49. The server computer of claim 39, wherein the set of instructions further comprise:
2 a plurality of instructions that, upon execution by the processor, cause the server computer
3 to renew the local copy of the first session state; and
4 a plurality of instructions that, upon execution by the processor, cause the server computer
5 to broadcast a command to other server computers of the group of server computers that, when
6 executed by an executing server computer, causes the executing server computer to locally renew
7 the first session state.

1 50. The server computer of claim 39, wherein the set of instructions further comprise:
2 a plurality of instructions that, upon execution by the processor, cause the server computer
3 to receive a request for a requested session state from a requesting server computer of the group of
4 server computers;
5 a plurality of instructions that, upon execution by the processor, cause the server computer
6 to access a local copy of the requested session state; and
7 a plurality of instructions that, upon execution by the processor, cause the server computer

to transmit a copy of the requested session state to the requesting server computer.

51. The server computer of claim 50, wherein:

the request is sent to the primary owner and secondary owner of the requested session state;

and

the server computer is the primary owner or secondary owner of the requested session state.

52. The server computer of claim 50, wherein the request is sent to all server computers

of the group of server computers.

53. The server computer of claim 39, wherein the set of instructions further comprise

a plurality of instructions that, upon execution by the processor, cause the server computer to identify at least one other server computer of the group of server computers.

54. The server computer of claim 39, wherein the set of instructions further comprise

a plurality of instructions that, upon execution by the processor, cause the server computer to identify a secondary owner of the other server computers of the group of server computers.

55. The server computer of claim 39, wherein the set of instructions further comprise

a plurality of instructions that, upon execution by the processor, cause the server computer to publish a plurality of locally stored session states to other server computers of the group of server computers.

1 56. The server computer of claim 55, wherein the server computer coordinates the
2 session states it publishes with session states published by other server computers of the group of
3 server computers.